

Solving Nonlinear Eigenvalue Problems using an Improved Newton Method

S.A Shahzadeh Fazeli
Parallel Processing Laboratory
Faculty of Mathematics
Yazd University
Yazd, P. O. Box 89195-741, Iran

F. Rabiei
Parallel Processing Laboratory
Faculty of Mathematics
Yazd University
Yazd, P. O. Box 89195-741, Iran

Abstract—Finding approximations to the eigenvalues of nonlinear eigenvalue problems is a common problem which arises from many complex applications. In this paper, iterative algorithms for finding approximations to the eigenvalues of nonlinear eigenvalue problems are verified. These algorithms use an efficient numerical approach for calculating the first and second derivatives of the determinant of the problem. Here we present and examine a technique for solving nonlinear eigenvalue problems using Newton method. Computational aspects of this approach for a nonlinear eigenvalue problem are analyzed. The efficiency of the algorithm is demonstrated using an example.

Keywords—nonlinear eigenvalue problems; Newton method; LU-decomposition; refined eigenvalues

I. INTRODUCTION

A method is presented in [5] for obtaining lower and upper bounds on eigenvalues and eigenfunctions for linear integral equations.

Another method is described in [6] for the calculation of the eigenvalues of general integral operators. Several classical results from functions of a complex variable and the theory of integral equations are combined with a recent technique for converting Fredholm integral equations into an initial-valued system of differential equations. The algorithm, which is based on a Cauchy system for the Fredholm determinants, is related to the Nyström method and results of Anselone and Atkinson become applicable.

Another method is presented in [7] for the calculation of the eigenvalues and eigenfunctions of complex-valued symmetric kernels which occur in laser theory. The method combines some classical results of integral equations and complex variables with a recent technique for transforming Fredholm integral equations into a Cauchy system of differential equations.

In this paper, a numerical method for solving the following eigenvalue problem is proposed. Here the method and the notations presented in [4] is used to simplify our method.

Let $D(\lambda)$ be a given n -by- n matrix that is a nonlinear function of the spectral parameter λ . It is required to find the values $\lambda \in C$ (called the eigenvalues) such that the equations

$$x^*D(\lambda) = 0, \quad D(\lambda)y = 0, \quad (1)$$

have nontrivial solutions $x, y \in C^n$.

Here, the asterisk in the superscript indicates the Hermitian adjoint operation. Both problems in (1) have the same desired values of λ that solve the equation

$$f(\lambda) \equiv \det D(\lambda) = 0. \quad (2)$$

In what follows, it is assumed that the entries of $D(\lambda)$ are sufficiently smooth functions of λ varying in a certain domain. This process is an improved Newton method as applied to finding a simple real eigenvalue considered as a root of the corresponding nonlinear scalar equation (2); however, in equation (2), the left-hand side is not expressed in an explicit form. Instead, it is proposed an algorithm for calculating the values of $f(\lambda), f'(\lambda)$ at a fixed λ ; to this end, the LU-decomposition of $D(\lambda)$ is used.

Moreover, the proposed algorithm, combined with the argument principle for analytic functions, makes it possible to find the number of eigenvalues belonging to a given domain G in the complex λ -plane, as well as to find initial approximations to all of these eigenvalues. The approximations found can then be refined using any of the available iterative methods; in particular, an improved Newton method can be applied.

II. CALCULATING $f(\lambda)$ AND $f'(\lambda)$

It is well known that, for any fixed λ , the matrix $D(\lambda)$ can be represented in the form

$$D(\lambda) = L(\lambda)U(\lambda), \quad (3)$$

where $L(\lambda)$ is a lower triangular matrix with unit diagonal and $U(\lambda)$ is an upper triangular matrix. It follows that

$$f(\lambda) = \det L(\lambda)\det U(\lambda) = \prod_{i=1}^n u_{ii}(\lambda),$$

Since the entries of the square matrix $D(\lambda)$ (and, hence, those of $U(\lambda)$)are differentiable functions of λ , that is

$$f'(\lambda) = \sum_{r=1}^n v_{rr}(\lambda) \prod_{i=1, i \neq r}^n u_{ii}(\lambda),$$

for any λ , here $v_{ii}(\lambda) = u'_{ii}(\lambda)$.

To find $v_{ii}(\lambda)$ equation (3) is differentiable with respect to λ . This yields

$$B(\lambda) = M(\lambda)U(\lambda) + L(\lambda)V(\lambda), \quad (4)$$

where

$$B(\lambda) = D'(\lambda), M(\lambda) = L'(\lambda), V(\lambda) = U'(\lambda),$$

and $v_{ii}(\lambda)$ are the entries in the matrix $V(\lambda)$. Thus, to calculate $f(\lambda), f'(\lambda)$, at a fixed $\lambda = \lambda_m$, it is necessary to find the matrices

$$\begin{aligned} D &= LU, \\ B &= MU + LV. \end{aligned} \quad (5)$$

This yields

$$\begin{aligned} f(\lambda_m) &= \prod_{i=1}^n u_{ii}(\lambda_m), \\ f'(\lambda_m) &= \sum_{r=1}^n v_{rr}(\lambda_m) \prod_{i=1, i \neq r}^n u_{ii}(\lambda_m). \end{aligned} \quad (6)$$

The entries of the matrices appearing in decomposition (5) can be calculated using the recursions

$$\begin{aligned} r &= 1, 2, \dots, n \\ u_{rk} &= d_{rk} - \sum_{j=1}^{r-1} l_{rj} u_{jk}, \\ l_{ir} &= [d_{ir} - \sum_{j=1}^{r-1} l_{ij} u_{jr}] u_{rr}^{-1}, \quad i = r + 1, \dots, n, \\ v_{rk} &= d_{rk} - \sum_{j=1}^{r-1} (m_{rj} u_{jk} + l_{rj} v_{jk}), \quad i = r + 1, \dots, n, \\ m_{ir} &= [d_{ir} - \sum_{j=1}^{r-1} (m_{ij} u_{jr} + l_{ij} v_{jr}) - l_{ir} v_{rr}] u_{rr}^{-1}, \\ & i = r + 1, \dots, n. \end{aligned}$$

This algorithm may be unstable and even incorrect if $u_{rr} = 0$ for some r . To avoid such occurrences, one uses permutations of the rows (and/or columns) of D in the process of its LU -decomposition; simultaneously, a pivot is chosen similarly to the Gaussian elimination. In this case, decomposition of (5) can be written as

$$\begin{aligned} PD &= LU, \\ PB &= MU + LV, \end{aligned}$$

where P is a permutation matrix; and $\det P = (-1)^q$, where q is the number of permutations. Thus, relations (6) take the form

$$\begin{aligned} f(\lambda_m) &= (-1)^q \prod_{i=1}^n u_{ii}(\lambda_m), \\ f'(\lambda_m) &= (-1)^q \sum_{r=1}^n v_{rr}(\lambda_m) \prod_{i=1, i \neq r}^n u_{ii}(\lambda_m). \end{aligned}$$

After that, this algorithm is used for calculating the derivatives on the basis of the LU -decomposition of $D(\lambda)$.

III. COMPUTATIONAL ASPECTS OF THE ALGORITHM

The argument principle was repeatedly used to solve various problems in which the number of eigenvalues belonging to a given domain must be determined. By assumption, the characteristic function (2) is analytic. Suppose that f has m zeros $\lambda_1, \dots, \lambda_m$ in G (with the multiplicities counted) and has no zeros on the boundary Γ of G .

It is well known from the argument principle that the number m is determined by the formula [1]

$$m = S_0 = \frac{1}{2\pi i} \int_{\Gamma} \frac{f'(\lambda)}{f(\lambda)} d\lambda. \quad (7)$$

Define the quantities

$$S_k = \frac{1}{2\pi i} \int_{\Gamma} \lambda^k \frac{f'(\lambda)}{f(\lambda)} d\lambda, \quad k = 0, 1, 2, \dots, \quad (8)$$

then, it can be shown that

$$\sum_{j=1}^m \lambda_j^k = S_k, \quad k = 0, 1, 2, \dots \quad (9)$$

Thus, if m and $S_k, (k = 1, 2, \dots, m)$ are known, then system (9) determines the zeros of function (2), that is, all the eigenvalues of problem (1) belonging to the given domain G .

Let us dwell on the computational aspects of this algorithm, namely, on the stage at which the quantities $S_k (k = 1, 2, \dots, m)$ are calculated. Thus, it is proposed to use the LU -decomposition of $D(\lambda)$ for calculating both $f(\lambda)$ and $f'(\lambda)$.

Without loss of generality, take the circle $G(\lambda^*, \rho)$ of radius ρ centered at λ^* as the domain G bounded by the contour Γ . The change of the variable

$$\lambda(t) = \lambda^* + \rho \exp(2\pi i t)$$

transforms integral (8) to the form

$$S_k = \int_0^1 \lambda(t)^k \rho \exp(2\pi i t) \frac{f'(\lambda(t))}{f(\lambda(t))} dt. \quad (10)$$

Partition the interval $[0, 1]$ into N equal subintervals and replace integral (10) by a quadrature (for instance, following [2], the rectangle rule can be used).

This yields

$$S_k = \frac{1}{N} \sum_{j=1}^N \lambda_j^k \rho \exp(i \frac{2\pi j}{N}) \frac{f'(\lambda_j)}{f(\lambda_j)}, \quad (11)$$

where

$$\lambda_j = \lambda^* + \rho \exp(i \frac{2\pi j}{N}).$$

Thus, formula (11) requires that only the values of $f(\lambda)$ and of its derivative on the boundary of G is calculated. This can be done by using decomposition (5). Then, using

representations (6), it can be rewritten the ratio $\frac{f'(\lambda)}{f(\lambda)}$ in the form

$$\frac{f'(\lambda_j)}{f(\lambda_j)} = \sum_{r=1}^n \frac{v_{rr}}{u_{rr}}. \quad (12)$$

In view of (12), the following formulas for calculating $S_k (k = 0, 1, \dots, m)$ is obtained:

$$S_k = \frac{1}{N} \sum_{j=1}^N (\lambda_j^k \rho \exp(i \frac{2\pi j}{N} \sum_{r=1}^n \frac{v_{rr}}{u_{rr}})). \quad (13)$$

Hence, using relations (13), the number $m = S_0$ of the eigenvalues belonging to G , as well as the right-hand side of equation (9) can be found. Following [4], the Newtons method for solving this system, may be applied which results in certain (in general, rather rough) approximations to all the eigenvalues are belonging to G .

IV. AN IMPROVED NEWTON METHOD(IM_NEWTON)

The use of iterative method for refining the rough approximations to the eigenvalues that were obtained by the algorithm described above is proposed.

Theorem 1: Let $p(x)$ be a real polynomial of degree $n \geq 2$, all zeros of which are real, $\xi_1 \geq \xi_2 \geq \dots \geq \xi_n$. Let α_1 be the largest zero of $p'(x): \xi_1 \geq \alpha_1 \geq \xi_2$. For $n = 2$, it is also required that $\xi_1 > \xi_2$. Then for every $z > \xi_1$, the numbers

$$z' = z - \frac{P(z)}{P'(z)},$$

$$y = z - 2 \frac{P(z)}{P'(z)},$$

$$y' = y - \frac{P(y)}{P'(y)}$$

are well defined and satisfy $\alpha_1 < y$, $\xi_1 \leq y' \leq z'$. It is readily verified that $n = 2$ and $\xi_1 = \xi_2$ imply $y = \xi_1$ for any $z > \xi_1$.

Suppose that an approximation λ_0 to the eigenvalue λ_* is given such that the Im_Newton method described in [3] can be initiated from λ_0 :

$$y_m = \lambda_m - 2 \frac{f(\lambda_m)}{f'(\lambda_m)}, \quad (14)$$

$$y'_{m+1} = y_m - \frac{f(y_m)}{f'(y_m)}. \quad (15)$$

At each step of iterative process (14), the values of $f(\lambda)$ and its derivatives at a specific λ are used. Therefore, to calculate these values, use of decomposition (5) and of relations (6) is made. As a result, process (14) and (15) takes the form

$$y_m = \lambda_m - 2 \left(\sum_{k=1}^n \frac{v_{kk}}{u_{kk}} \right)^{-1}, \quad (16)$$

$$y'_{m+1} = y_m - \left(\sum_{k=1}^n \frac{v_{kk}}{u_{kk}} \right)^{-1}. \quad (17)$$

Thus, the following algorithm for solving the nonlinear eigenvalue problem (1) is proposed.

Algorithm 1: Iterative process for refining the rough approximations

Step 1. Choose an initial approximation λ_* to the s th eigenvalue of problem (1).

Step 2. for $m = 0, 1, \dots$ until the required accuracy is attained do.

Step 3. Determine the entries u_{kk}, v_{kk} in decomposition (5).

Step 4. Calculate y_m and y'_{m+1} using formula (16) and (17).

Step 5. Let $\lambda_{m+1} = y'_{m+1}$.

Step 6. end for.

V. NUMERICAL EXAMPLE

The algorithm for calculating rough approximates of eigenvalues presented in section II was tested for the quadratic eigenvalue problem with the matrix

$$D(\lambda) = \lambda^2 A_0 + \lambda A_1 + A_2,$$

where

$$A_0 = \begin{bmatrix} 1 & 0.17 & -0.25 & 0.54 \\ 0.47 & 1 & 0.67 & -0.32 \\ -0.11 & 0.35 & 1 & -0.74 \\ 0.55 & 0.43 & 0.36 & 1 \end{bmatrix},$$

$$A_1 = \begin{bmatrix} 0.22 & 0.02 & 0.12 & 0.14 \\ 0.02 & 0.14 & 0.04 & -0.06 \\ 0.12 & 0.04 & 0.28 & 0.08 \\ 0.14 & -0.06 & 0.08 & 0.26 \end{bmatrix},$$

$$A_2 = \begin{bmatrix} -3.0475 & -2.1879 & -1.9449 & -2.8242 \\ -2.6500 & -2.4724 & -2.3515 & -2.1053 \\ -0.7456 & -0.6423 & -1.3117 & -0.1852 \\ -4.0500 & -3.0631 & -2.8121 & -3.7794 \end{bmatrix}.$$

First, the number of eigenvalues belonging to a given domain G was determined. In presented calculations, G was a circle $G(\lambda^*, \rho)$ centered at $\lambda^* = 0$.

The following radii were used: $\rho = 0.3, 1.0, 1.3$, and 3.0 . The results are presented in TABLE I. For each ρ , the number of eigenvalues belonging to G , the eigenvalues themselves (λ), and the number k of iteration steps required to calculate the eigenvalues to an accuracy of $\epsilon = 10^{-4}$ are shown.

The approximate eigenvalues found at this stage may not have the required accuracy. To refine these approximations, the

TABLE I. THE NUMBER OF EIGENVALUES

$\rho = 0.3$ $k = 2$	$\rho = 1.0$ $k = 68$	$\rho = 1.3$ $k = 25$	$\rho = 3.0$ $k = 77$
λ	λ	λ	λ
0.2423	0.6383	0.2383	2.3229
	0.2423	-0.8499	0.7967
	-0.3778	-1.2261	0.6383
	0.7967	0.3878	0.2423
	-0.8394	0.7956	-0.3778
		0.6408	-0.8394
			-1.2234
			-2.6354

[7] M. R. Scott and R. E. Kalaba, "An Initial Value Method for Integral Operators: IV -Complex-Valued Kernels of Laser Theory," J. Quant. Spec. Rad. Trans., 13, pp. 509-515 (1973).

algorithms presented in above can be used.

The results of refine are presented in TABLE II. The exact eigenvalues, and initial approximations, are respectively shown in columns 2 and 3. The next column provides numbers of iterations steps required to calculate the eigenvalues to an accuracy of $\epsilon = 10^{-9}$ and the refined eigenvalues are presented in the last column.

TABLE II. REFINED EIGENVALUES

k	<i>exact eigen.</i>	<i>initial approx.</i>	m	<i>refined eigen.</i>
1	0.2422606951	0.2423	2	0.2422606954
2	0.6382838292	0.6383	3	0.6382838295
3	0.7967066727	0.6383	3	0.7967066725
4	2.322748800	2.3229	2	2.322748800
5	-0.777442689	-0.3778	2	-0.777442685
6	-0.8393977662	-0.8394	1	-0.8393977663
7	-1.223471197	-1.2234	2	-1.223471198
8	-2.635389128	-2.6354	2	-2.635389128

VI. CONCLUSION

In this paper, an improved Newton method for solving non-linear eigenvalue problems was presented. The results showed that the improved Newton method is an efficient method. The presented method is very efficient and competitive with other methods used to solve nonlinear eigenvalue problems. Here MATLAB 2012 software was used to implement the algorithm.

REFERENCES

[1] A. V. Bitsadze, Fundamentals of the Theory of Analytical Functions of a Complex Variable (Nauka, Moscow, 1972) [in Russian].
 [2] S. V. Kartyshev, Numerical Method for Solving the Eigenvalue Problem for Sparse Matrices Depending Nonlinearly on a Spectral Parameter, Zh. Vychisl. Mat. Mat. Fiz. 29, pp. 1898-1903 (1989).
 [3] A. Melman, The double-step Newton method for polynomials with all real zeros, Department of Applied Mathematics, School of Engineering, Santa Clara University, Santa Clara, CA 95053, United States.
 [4] B. M. Podlevskii, On Certain Two-Sided Analogues of Newtons Method for Solving Nonlinear Eigenvalue Problems equations, Institute of Applied Mathematics and Mechanics, National Academy of Sciences of Ukraine, ul. Nauchnaya 3-b, Lviv, 79000 Ukraine.
 [5] M. R. Scott and J. W. Burgmeier, "A Method for Obtaining Bounds on Eigenvalues and Eigenfunctions by Solving Non-Homogeneous Integral Equations," Computing, 10, pp. 3-22 (1972).
 [6] M. R. Scott and R. E. Kalaba, "An Initial Value Method for Integral Operators: I -Complex Eigenvalues," J. Comp. Phys., 12, pp. 364-3648 (1973).